

PRODUCT REQUIREMENTS REVIEW PROCEDURE (FOR LOW CONTROL)

- Objectives:**
- Define the process to review Product Requirements (e.g., System Requirements, Software Requirements) and changes to Product Requirements.
 - Establish a standard set of Product Requirements quality criteria (documented in the Product Requirements Checklist in Appendix A).
 - Ensure that Product Requirements have undergone a thorough review and update using the Product Requirements Checklist.
 - Ensure that Product Requirements are complete, consistent, correct, unambiguous, and verifiable.

Scope: This procedure applies to the development and review of technical Product Requirements for LaRC Low-Control projects/activities (as opposed to requirements defined in a Statement of Work which are reviewed using LMS-CP-5523). A project/activity is defined as Low Control if any one of the following is true; the project/activity has:

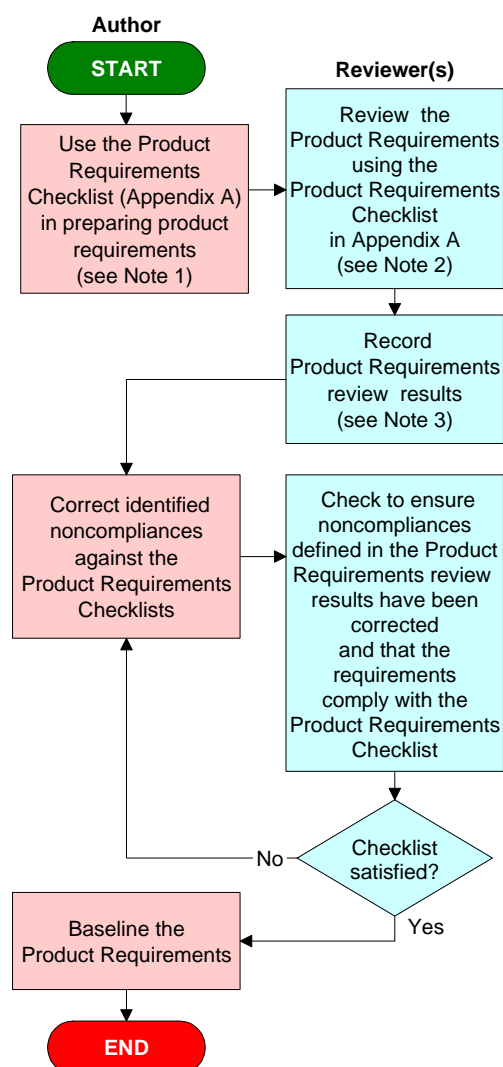
- a total potential personnel resource investment of less than 8 FTE (but more than 2 FTE) across all lifecycle phases; depending on the full cost of the FTE, this may be between \$0.5M to over \$2M,
- a potential for waste of facility resources (e.g., electricity, liquid nitrogen, operational costs) of less than \$250K – but greater than \$50K if failure occurred; this is only the cost of potential waste of facility resources if failure occurred – not the total cost of the project/activity,
- potential risk to routine operations of at most a facility inconvenience if a failure occurred (e.g., a facility test has to be re-run due to the failure), or
- potential for knowledge of a failure at the Directorate or Branch level if failure occurred.

If the project/activity 1) exceeds any one of the above; or 2) has the potential for safety mishap, i.e., loss of life, serious injury, or damage to equipment greater than \$1M; or 3) has the potential for catastrophic or partial mission failure, it should follow LMS-CP-5526 "Product Requirements Development and Management". (For a more extensive definition of Control Class, see Appendix D of LMS-CP-5526). Mentoring and answers to questions on this procedure can be obtained by calling the procedure owner and will be provided on an as-requested basis.

Records generated:

- Evidence that requirements review issues were resolved
- Updated Product Requirements

Approval: original signed on file Date 7/21/06
Associate Director for Operations

**Inputs/Entry Criteria:**

- The selected reviewer has experience with the subject matter of the Product Requirements.

Note 1:

For Product Requirements guidance and examples, see URL http://sw-eng.larc.nasa.gov/docs/requirements_capture_and_management.html.

Note 2:

Although the review using the Product Requirements Checklist can be conducted by a single reviewer, to ensure the Product Requirements are complete and correct, it is recommended that the requirements be reviewed by all stakeholders (e.g., customers, systems engineers, software engineers, hardware engineers).

Note 3:

Include the following in the Product Requirements review results:

1. Title of reviewed Product Requirements document
2. Name of reviewer
3. Date of review
4. Review comments—Include the following for each review comment:
 - a. Identification of Product Requirements paragraph(s) / sentence(s) to which the comment applies
 - b. Documentation of noncompliance against the Product Requirements Checklist (Appendix A)
 - c. Suggested correction (optional)

Outputs/Exit Criteria:

- The reviewer(s) has recorded the Product Requirements review results.
- The author has updated the Product Requirements in accordance with the reviewer's comments so that the updated document satisfies the criteria in the Product Requirements Checklist.
- The author retains evidence that the Product Requirements review issues were resolved.
- The Product Requirements have been baselined.
(Baseline: A specification or product that has been formally reviewed and agreed upon that thereafter serves as the basis for further development, and that can be changed only through change control procedures. [Modified from IEEE-610.12, IEEE Standard Glossary of Software Engineering Terminology])

Appendix A: Product Requirements Checklist

Instructions:

- This checklist only applies to Product Requirements.
- Statement of Work (SOW) contractor activity or service requirements are reviewed using the procedure LMS-CP-5523. For contracted efforts, the government requirements are part of the contractual binding language. Product Requirements can be included as part of the Statement of Work (SOW); they do not have to be in a separate document.

Note: Items in **gray** text are provided as examples and explanatory guidance.

A.1 Use of Correct Terms

- **Shall** = requirement
- **Should** = goal
- **Will** = facts or declaration of purpose
- NEVER use '**must**'

A.2 Editorial Checklist

- Is the requirement in the form "product ABC shall XYZ". E.g., "The product shall (do, perform, provide, weigh, or other verb followed by a description of what)."
Example Product Requirements:
 - The system shall operate at a power level of...
 - The software shall acquire data from the...
 - The structure shall withstand loads of...
 - The hardware shall have a mass of...
- Does the requirement use active rather than passive voice? Passive voice leads to vague statements. (For example, state: "The system shall perform nightly backups", instead of "nightly backups shall be performed.")
- Is the requirement stated positively as opposed to negatively? (I.e., replace statements such as: "The structure shall not exceed the standards specified..." with "The structure shall comply with the standards specified...")
- Is the requirement grammatically correct?
- Is the requirement free of typos, misspellings, and punctuation errors?
- Does the requirement comply with the project's template and style rules, if one exists?
- Have all acronyms been defined in an Acronyms List or spelled out in the first occurrence?

A.3 Goodness Checklist

Is each requirement ...

- Free from indefinite pronouns ("this", "that", "these", "those") without clear antecedents? (E.g., replace statements such as "These shall be in balance." with "The fan blades shall be in balance.")
- A standalone statement that expresses only one thought? (As opposed to multiple requirements in a single statement.) Compound sentences that contain multiple requirements in a single statement need to be split into multiple simple sentences. For example, "The box shall be lockable and have a slot on top." should be rewritten as: "The box shall be lockable. The box shall have a slot on top."
- Located in the proper section of the document? Is the text compatible with the title of the section it's under? Are sub-headings compatible with the subject matter of a heading?
- Have all redundant requirements been removed? Redundant requirements (i.e., a requirement that appears in more than one place) can reduce clarity, increase ambiguity, and lead to contradictions.

A.4 Content Review / Inspection Checklist

CLARITY

1. Are the requirements clear? (I.e., are there aspects of the requirement that are not understood; can the requirement only be understood one way; can the requirement be misinterpreted?)
2. Are the requirements free of ambiguities? (I.e., free of terms like: "as appropriate", "etc.", "and/or", "support", "but not limited to", "be able to", "be capable of".)
3. Are the requirements concise and simple?

COMPLETENESS

1. Are requirements stated as completely as possible? Are the requirements free of “To Be Determined” (TBD) values? Have all incomplete requirements been captured as TBRs? (A best guess, marked “To Be Resolved” (TBR), with an explanation of what needs to be resolved, by whom, and when.)
2. Have all the engineering requirements been included? (E.g., systems, software, hardware requirements.)
3. Are any requirements missing? For example, have any of the following requirements areas been overlooked: functional, performance, interface, environment (development, manufacturing, test, transport, storage, operations), facility (manufacturing, test, storage, operations), transportation (among areas for manufacturing, assembling, delivery points, within storage facilities, loading), training, personnel, operability, safety, security, appearance and physical characteristics, and design.
4. Have all assumptions been explicitly stated? Have they been clearly marked as assumptions?
5. Are the requirements accompanied by intelligible rationale?
6. Is the rationale clearly labeled as such and in separate, stand alone paragraphs, not mixed with the requirements? When a Product Requirements document accompanies a contract or task order, the rationale is not part of the contractual binding language. The Product Requirements document and rationale may accompany the Request for Proposals.

COMPLIANCE

1. Are all requirements at the correct level? (E.g., system, segment, element, subsystem.) Too low a level of detail about the sub-elements can result in constraining the product design.
2. Are requirements specified in an implementation-free way so as not to obscure the actual requirements? Requirements should state WHAT is needed, NOT HOW to provide it, i.e., state the problem, not the solution. Ask, “Why do you need the “how” requirement?” The answer may point to the real “what” requirement.
3. Are requirements specified free of descriptions of operations? Operations and product requirements should be kept separate (e.g., in separate sections or documents). To distinguish between operations and product requirements ask the questions: “Does the developer have control over this? Is this a need the product must satisfy or an activity involving the product?” Sentences like “The operator shall...” are almost always operational statements not product requirements.

CONSISTENCY

1. Are the requirements stated consistently, without contradicting themselves or the requirements of related systems?
2. Is the terminology consistent with the user’s and sponsor’s terminology? With the project glossary?
3. Is the terminology consistently used throughout the document? E.g., does the requirement use consistent terminology to refer to the product and its lower-level entities?
4. Are the key terms included in the project’s glossary?

TRACEABILITY

1. Are all requirements needed? Is each requirement necessary to meet the parent requirement? Is each requirement a needed function or characteristic? Distinguish between needs and wants. If it is not necessary, it is not a requirement. Ask, “What is the worst that could happen if the requirement was not included?”
2. Are all requirements (functions, structures, and constraints) traced to the requirements at the level above; or, if this is the top-level requirements, are they traced to the Scope (i.e., need(s), goals, objectives, constraints, or operational concept)?
3. Is each requirement stated in such a manner that it can be uniquely referenced in subordinate documents?
4. Is the requirement allocation to the next lower level documented (e.g., traceability matrix)?

CORRECTNESS

1. Is each requirement correct?
2. Is each stated assumption correct? Has it been validated? Assumptions must be confirmed or marked as TBRs before the document can be baselined.
3. Are the requirements technically feasible?

FUNCTIONALITY

1. Are all described functions necessary and together sufficient to meet mission/system/project goals and objectives?

PERFORMANCE

1. Are all required performance specifications and margins listed? (E.g., consider timing, throughput, storage size, latency, accuracy and precision.)

2. Are all Product Specification Standards listed and are tolerances defined if needed? Product Specification Standards define what degree of adherence to the Product Requirements is good enough. (E.g., “mean time between failure less than”, “data capacity greater than or equal to”, “processor throughput plus or minus”.) For more information and examples on Product Specification Standards, see *Guidance on System and Software Metrics for Performance-Based Contracting* at: http://sw-eng.larc.nasa.gov/docs/statements_of_work.html.
3. Is each performance requirement realistic? Are the tolerances overly tight? Are the tolerances defensible and cost-effective? Ask, “What is the worst thing that could happen if the tolerance was doubled or tripled?”

INTERFACES

1. Are all external interfaces clearly defined?
2. Are all internal interfaces clearly defined?
3. Are all interfaces necessary, sufficient, and consistent with each other?

MAINTAINABILITY

1. Have the requirements for system maintainability been specified in a measurable, verifiable manner?
2. Are requirements written so that ripple effects from changes are minimized? (I.e., are requirements weakly coupled so that a change to one requirement will have a minimal influence on the other requirements?)

RELIABILITY

1. Are all necessary reliability requirements clearly defined, measurable, and verifiable?
2. Are there error detection, reporting, handling, and recovery requirements?
3. Are undesired events considered and their required responses specified? (Examples of undesired events are single event upset, data loss or scrambling, operator error.)
4. Have assumptions about the intended sequence of functions been stated? Are these sequences required?
5. Do these requirements adequately address the survivability after a software or hardware fault of the system from the point of view of hardware, software, operations personnel, and procedures?

VERIFIABILITY / TESTABILITY

1. Is a verification method(s) identified for each requirement? (I.e., test, demonstration, analysis, and/or inspection.) If a requirement is unverifiable, it must be rewritten. In most cases, the contractor is required to document the verification methods (e.g., verification matrix) and provide them to the government as a deliverable. There may be some exceptions to this, such as specific verification methods that the government may have to impose (e.g., federal regulations, national consensus codes, national and international standards requiring explicit test or inspection methods). The government requirements are part of the contractual binding language. If the effort will be performed in-house (not on contract), for each requirement, is the verification method(s) documented? Does a means exist to measure its accomplishment? Can you state the criteria required for verification? Can compliance be verified?
2. Are the requirements stated precisely to facilitate specification of (system, software, and hardware) test success criteria and requirements?
3. Are the requirements free of unverifiable terms? For example, eliminate phrases like: “flexible”, “easy”, “sufficient”, “safe”, “ad hoc”, “adequate”, “accommodate”, “user-friendly”, “useable”, “when required”, “if required”, “appropriate”, “fast”, “portable”, “light-weight”, “small”, “large”, “maximum”, “minimum”, “robust”, “quickly”, “easily”, “clearly”, other “ly” words, other “ize” words.

DATA USAGE

1. Where applicable, are “don’t care” conditions truly “don’t care”? Are “don’t care” condition values explicitly stated? Correct identification of “don’t care” values may improve a design’s portability. (“Don’t care” values identify cases when the value of a condition or flag is irrelevant, even though the value may be important for other cases.)